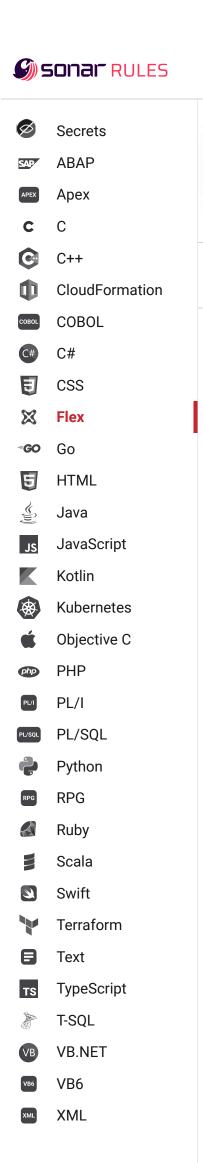
Q

Code Smell 61

Search by name...





Flex static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your FLEX code

Il rules 76	Rug 9 P Security Hotspot
	Tags
Security.allowDomain() should only be used in a tightly focused manner Culprise Vulnerability	Flex parser failure
The flash.system.Security.exactSettings property should never be set to false Uulnerability	When the Flex parser fails, it is possible to the file. This way, not only it is possible to parse but also to easily find out why they
Dynamic classes should not be used Code Smell	Available In: sonarcloud & sonarqube
"LocalConnection" should be configured to narrowly specify the domains with which local connections to other Flex application are allowed Vulnerability	© 2008-2022 SonarSource S.A., Switzerland. SONAR, SONARSOURCE, SONARLINT, SON trademarks of SonarSource S.A. All other tr property of their respective owners. All Privacy Poli
"default" clauses should be first or last Code Smell	<u>asy </u>
Event types should be defined in netadata tags	
Code Smell	
Event names should not be hardcoded in event listeners	
Code Smell	
The special "star" type should not be used	
Code Smell	
Variables of the "Object" type should not be used	
Code Smell	
Methods should not be empty	
Code Smell Code Smell	
Constant names should comply with a naming convention	
Code Smell Code Smell	

All branches in a conditional structure should not have exactly the same

Classes that extend "Event" should

implementation

Rug Bug

Analyze your code suspicious sible to record the failure as a violation on sible to track the number of files that do not y they do not parse. ıbe

zerland. All content is copyright protected. NT, SONARQUBE and SONARCLOUD are other trademarks and copyrights are the ers. All rights are expressly reserved. acy Policy

override "Event.clone()" • Bug
Constructors should not dispatch events
Rug
"ManagedEvents" tags should have companion "Event" tags
₩ Bug
Objects should not be instantiated inside a loop
Two branches in a conditional structure should not have exactly the same implementation
Constructor bodies should be as lightweight as possible
Only "while", "do" and "for" statements should be labelled
Statements, operators and keywords specific to ActionScript 2 should not be used
"for" loop stop conditions should be invariant
Unused function parameters should be removed
Code Smell